README:

I am using python2.7

When we extract the folder in a given location then there will be one folder called code will we formed in that code folder there will be aonther folder called search where all the codes will be present.

While running these commands we have to make sure that the pwd is in search folder.

And we must have tkinter on our system.

Q1:

First we have to test that the SearchAgent is working correctly by running:

    python2 pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch

We can test the algorithm by runnigng these commands … for these commands it will quickly find the soln if algo is correct

    python2 pacman.py -l tinyMaze -p SearchAgent

    python2 pacman.py -l mediumMaze -p SearchAgent

    python2 pacman.py -l bigMaze -z .5 -p SearchAgent

Q2:

Test your code the same way you did for depth-first search.

    python2 pacman.py -l mediumMaze -p SearchAgent -a fn=bfs

    python2 pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5

Q3:

python2 pacman.py -l mediumMaze -p SearchAgent -a fn=ucs

python2 pacman.py -l mediumDottedMaze -p StayEastSearchAgent

python2 pacman.py -l mediumScaryMaze -p StayWestSearchAgent

Q4:

we can test your A* implementation on the original problem of finding a path through a maze to a fixed position using the Manhattan distance heuristic (implemented already as manhattanHeuristic in searchAgents.py).

 python2 pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic

Q5:

python2 pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem

python2 pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem

Q6:

Implement a non-trivial, consistent heuristic for the CornersProblem in cornersHeuristic.

python2 pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5

Q7:

python2 pacman.py -l testSearch -p AStarFoodSearchAgent